

Atmospheric Chemistry in AM3

Larry Horowitz

(Larry.Horowitz@noaa.gov)

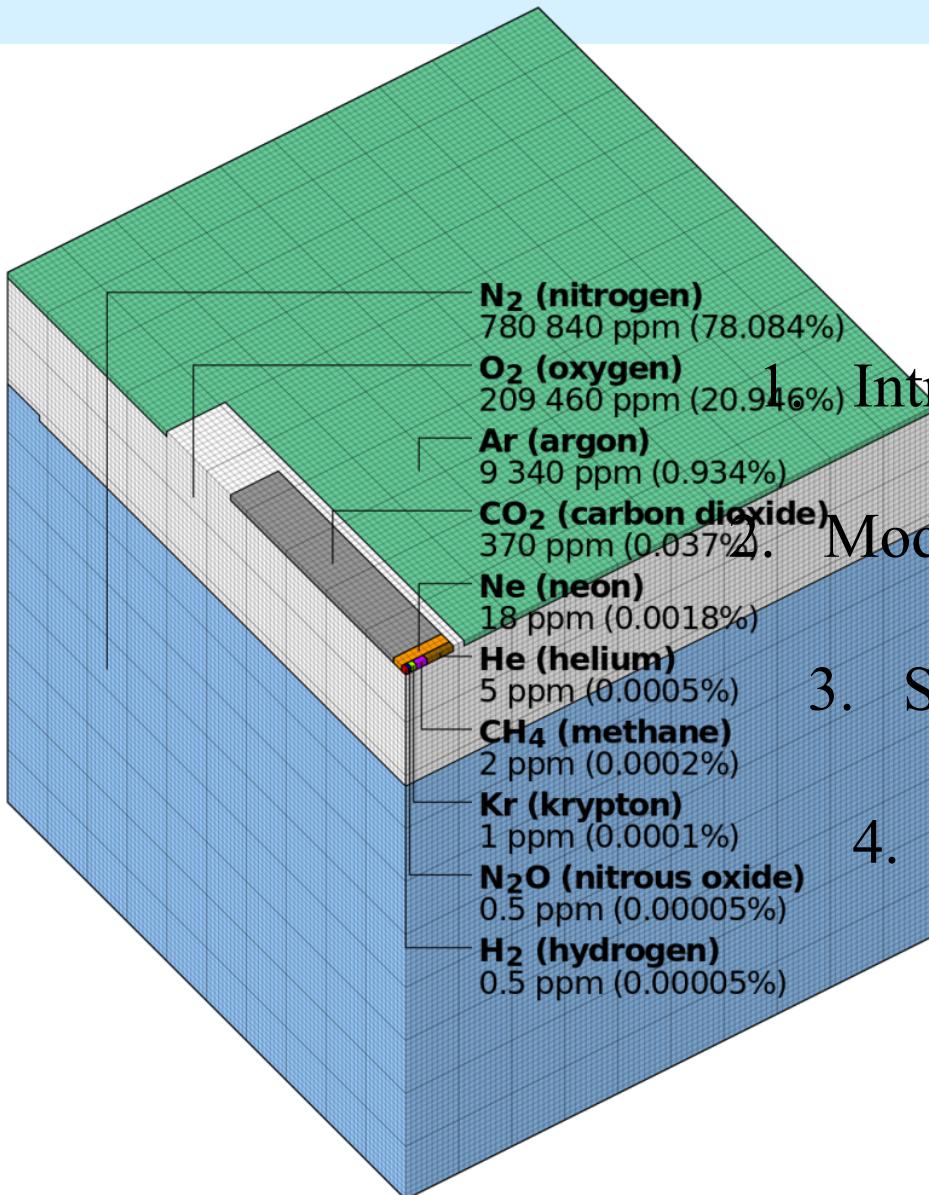
GFDL/NOAA, Princeton University

GFDL Summer School on Atmospheric Modeling

July 19, 2012



Atmospheric Chemistry



Outline

1. Introduction to atmospheric chemistry
2. Modeling atmospheric chemistry (AM3)

3. Some chemistry results from AM3
4. Configuring chemistry in AM3

Ozone 0.01-10 ppm
Nitrogen oxides ~ ppb
OH ~ ppt

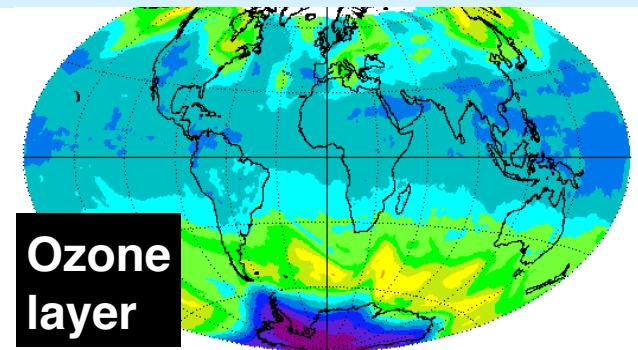
Big Issues in Atmospheric Chemistry



Disasters



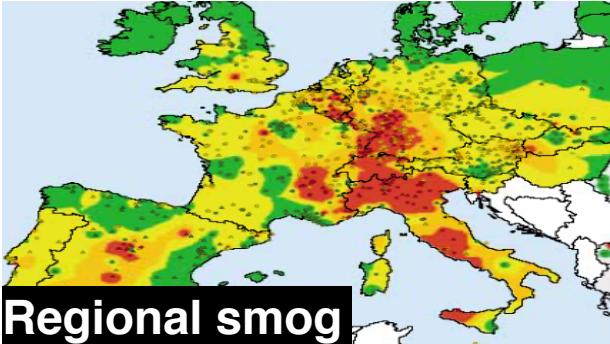
Visibility



Ozone
layer



Urban smog



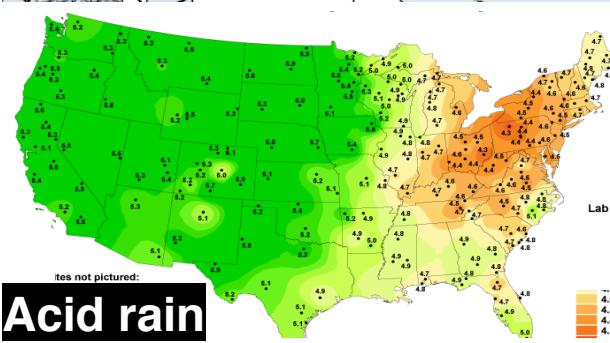
Regional smog



Climate



Point source



Acid rain



Biogeochemical cycles

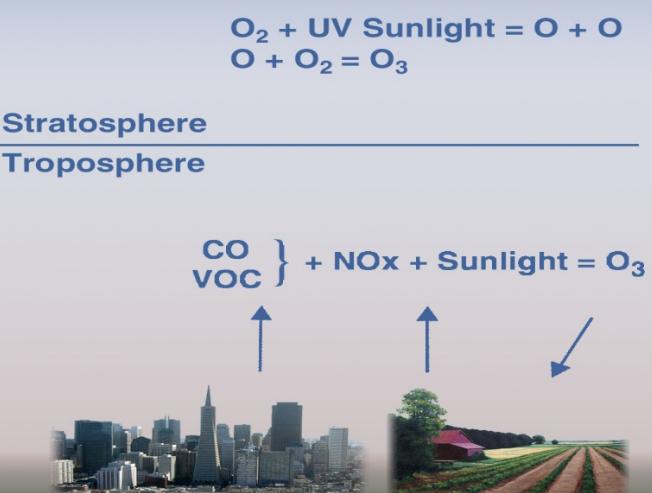
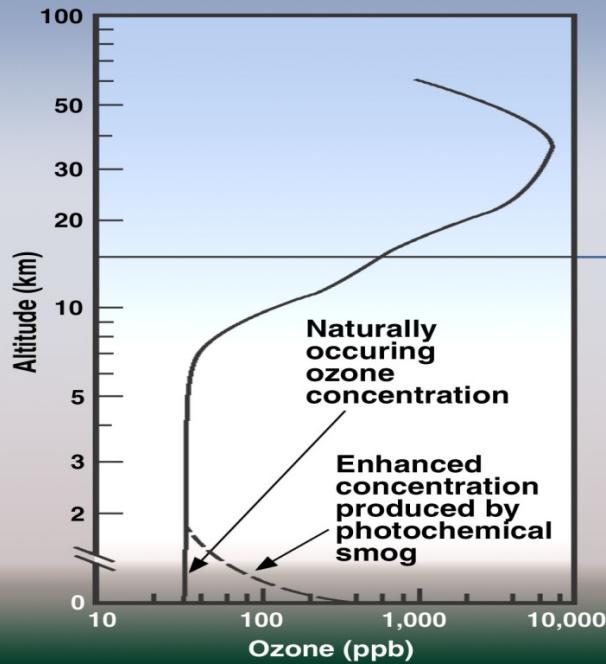
LOCAL
 $< 100 \text{ km}$

REGIONAL
 $100\text{-}1000 \text{ km}$

GLOBAL
 $> 1000 \text{ km}$
Daniel Jacob

Ozone: “Good Up High, Bad Nearby”

Good (UV shield) →
Bad (greenhouse gas) →
Good (OH source) →
Bad (smog) →

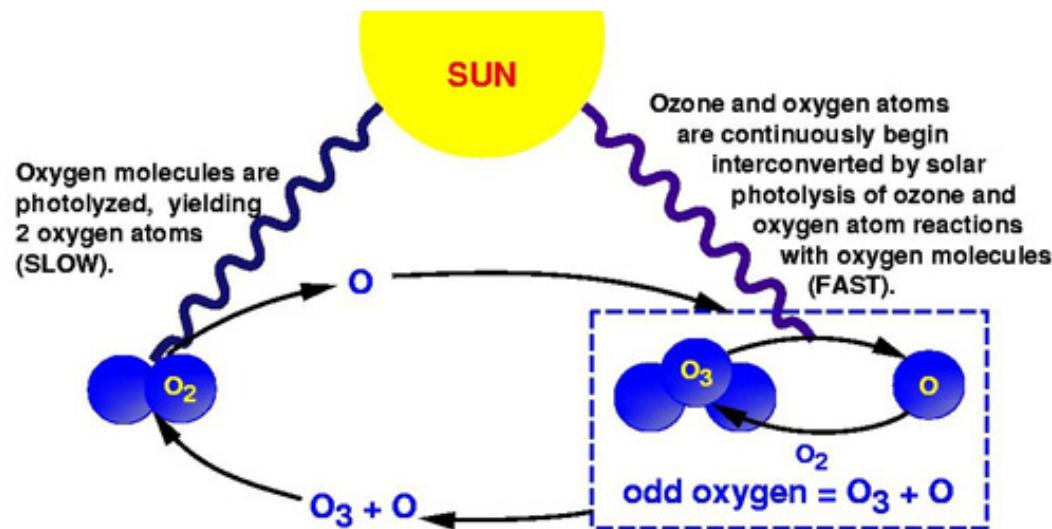


Tropospheric ozone precursors {

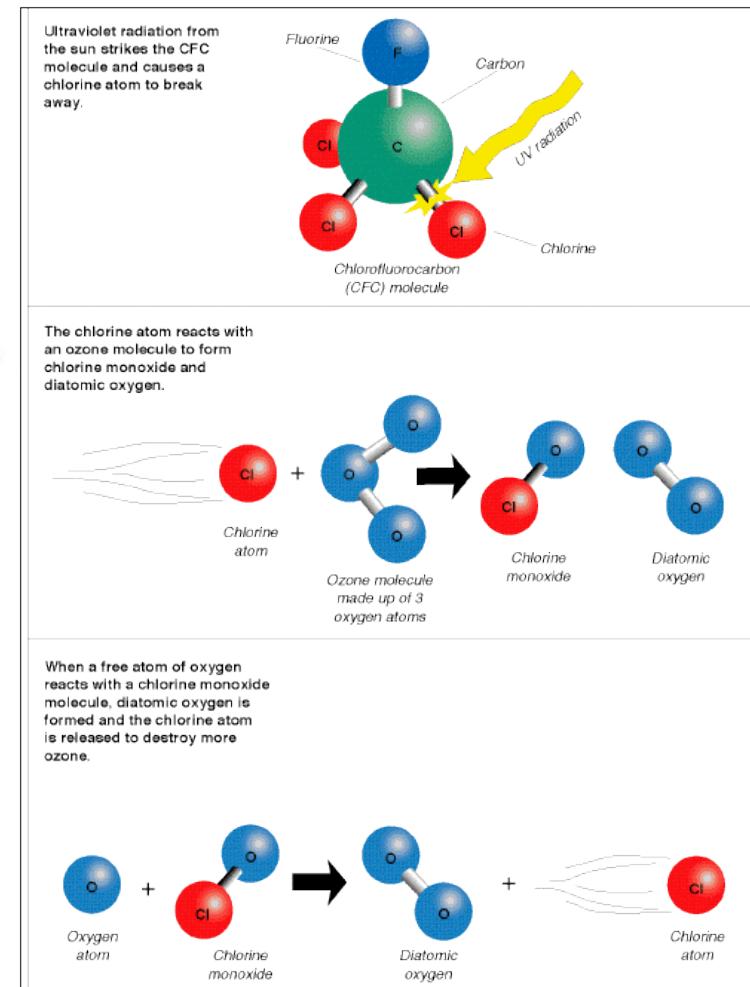
- Nitrogen oxide radicals; $\text{NO}_x = \text{NO} + \text{NO}_2$
Sources: combustion, soils, lightning
- Volatile organic compounds (VOCs)
 - Methane
Sources: wetlands, livestock, natural gas...
 - Non-methane VOCs (NMVOCs)
Sources: vegetation, combustion
- Carbon monoxide (CO)
Sources: combustion, VOC oxidation

Stratospheric Chemistry

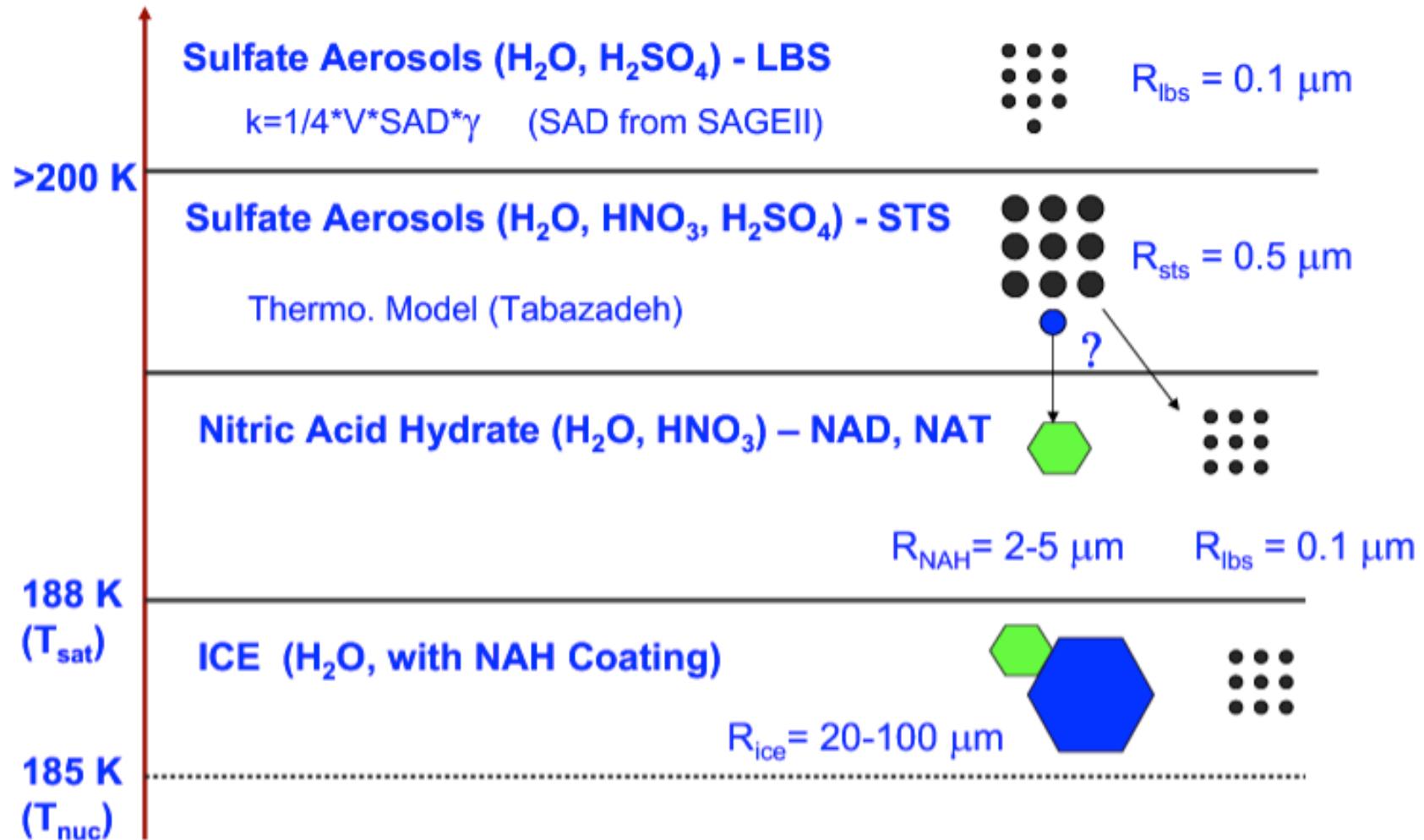
Natural Ozone Cycling



CFC-catalyzed ozone loss

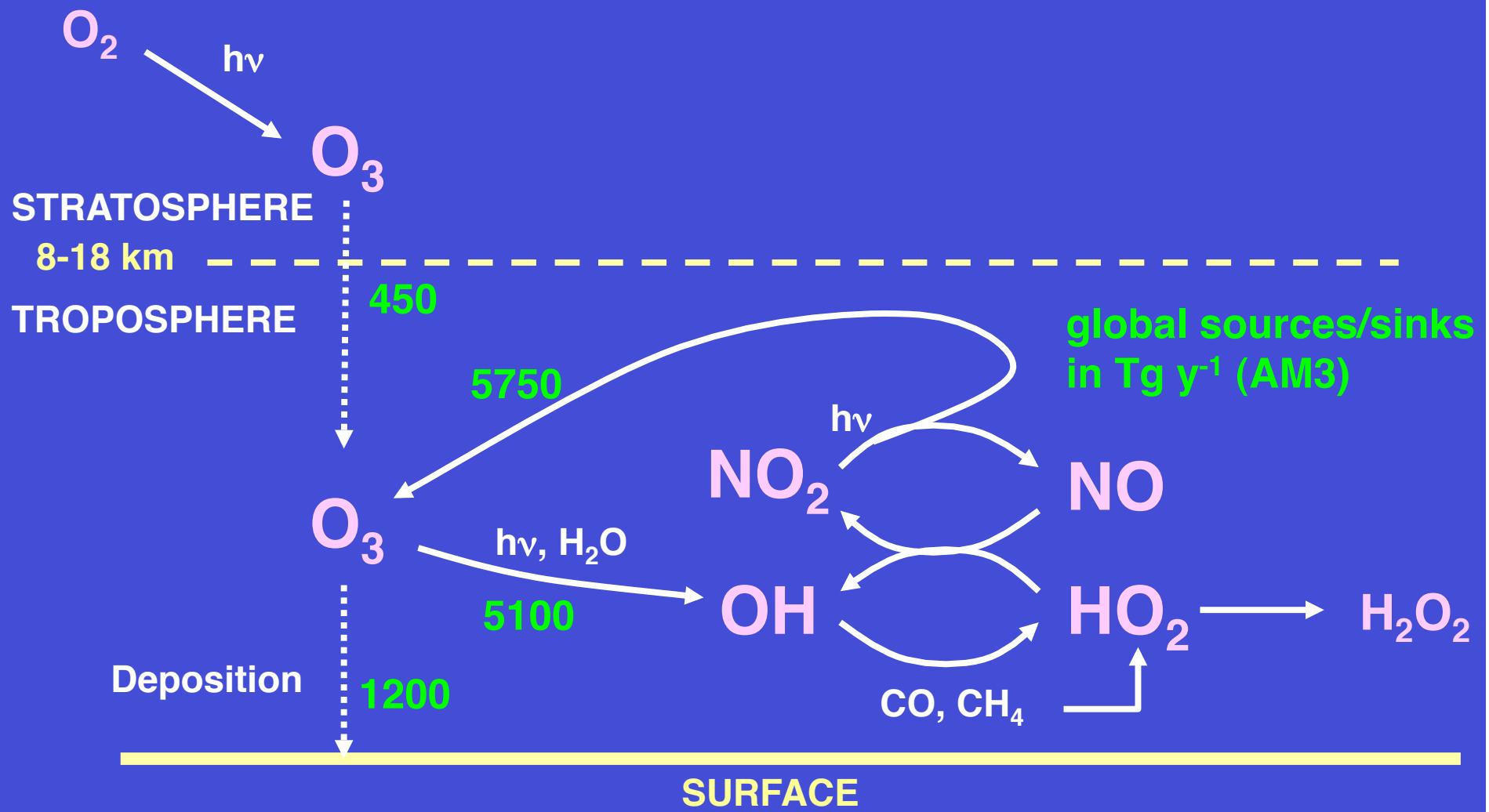


Heterogeneous Chemistry in the Stratosphere



Guy Brasseur

Tropospheric Chemistry Schematic

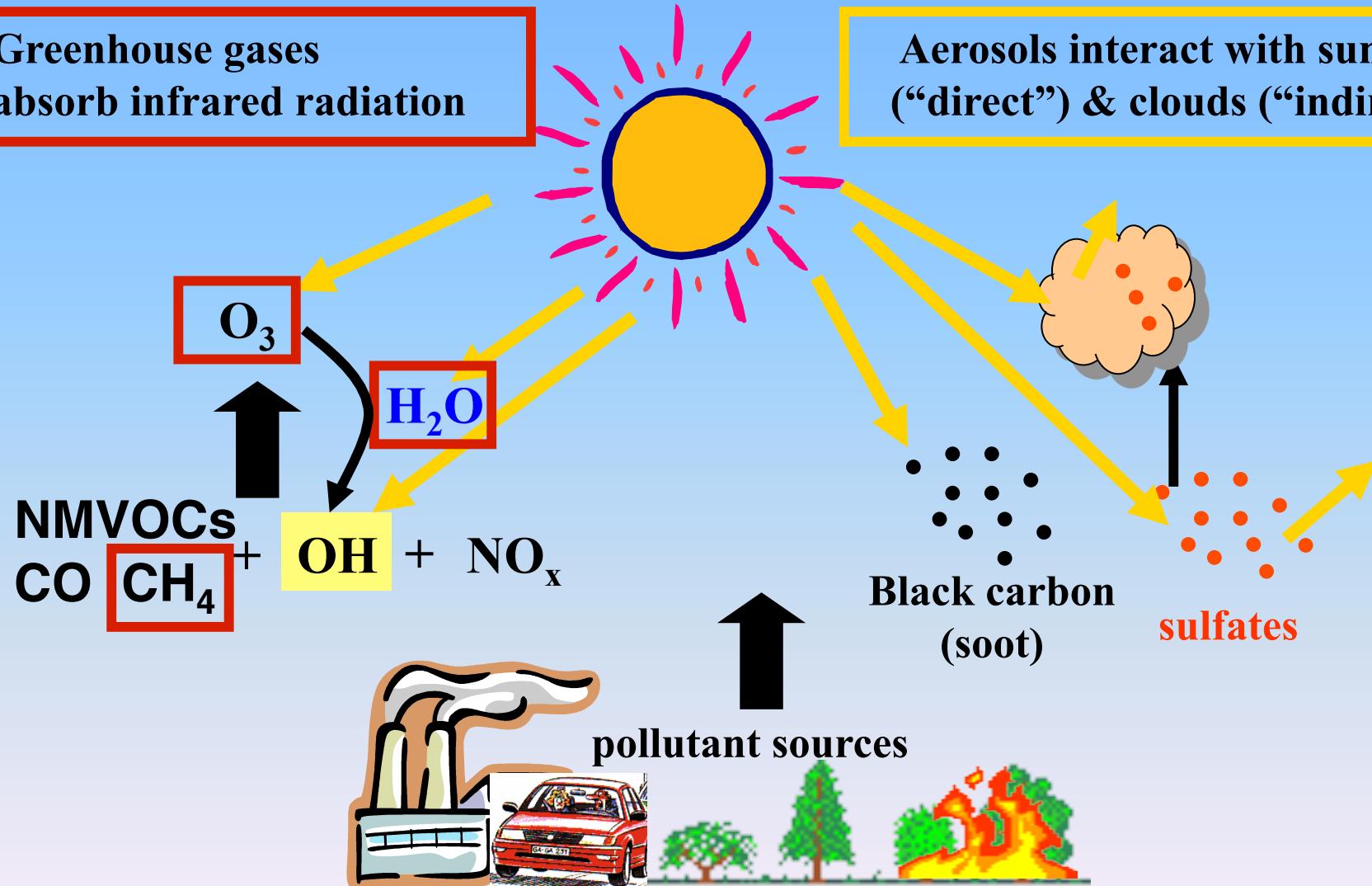




Air Pollutants Affect Climate

Greenhouse gases
absorb infrared radiation

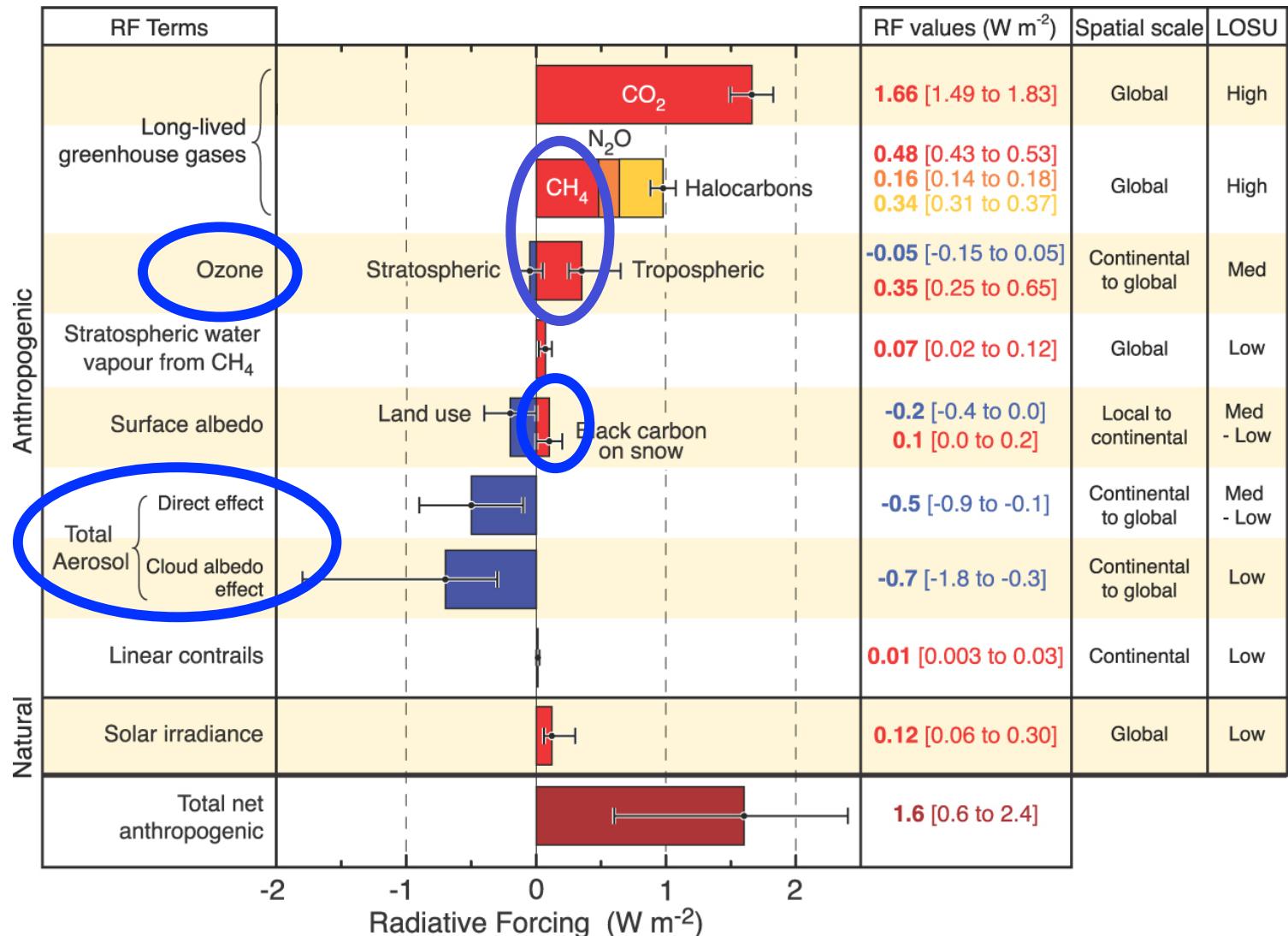
Aerosols interact with sunlight
("direct") & clouds ("indirect")



Surface of the Earth

A.M. Fiore

Radiative Forcing of Climate (1750 to present)



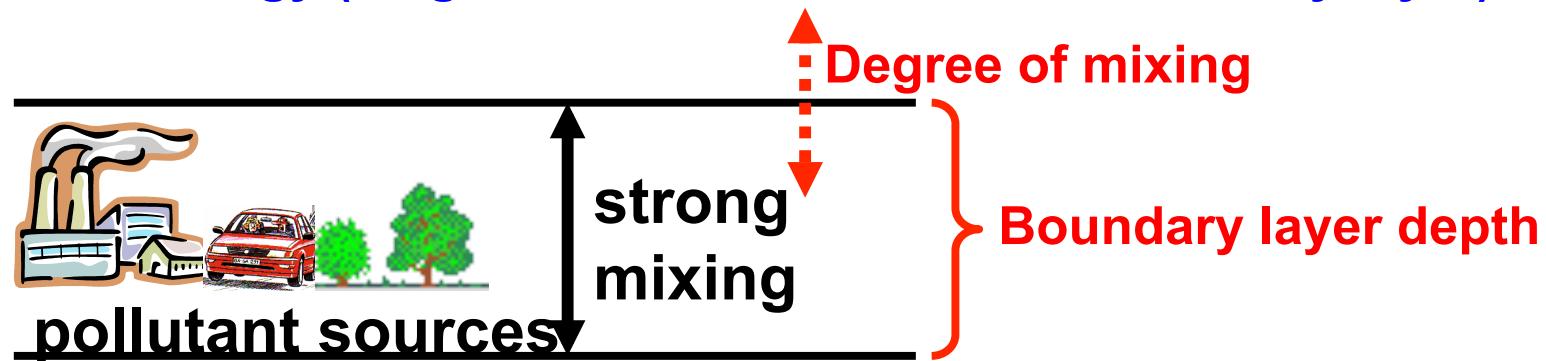
©IPCC 2007: WG1-AR4

Important contributions from air pollutants

IPCC [2007]

Climate (change) affects chemistry and air quality

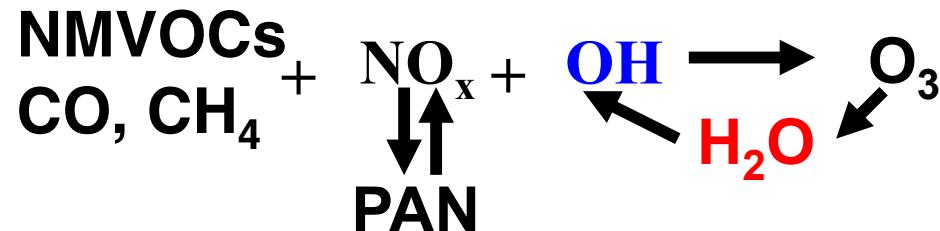
(1) Meteorology (stagnation vs. well-ventilated boundary layer)



(2) Emissions (biogenic, lightning NO_x, fires)

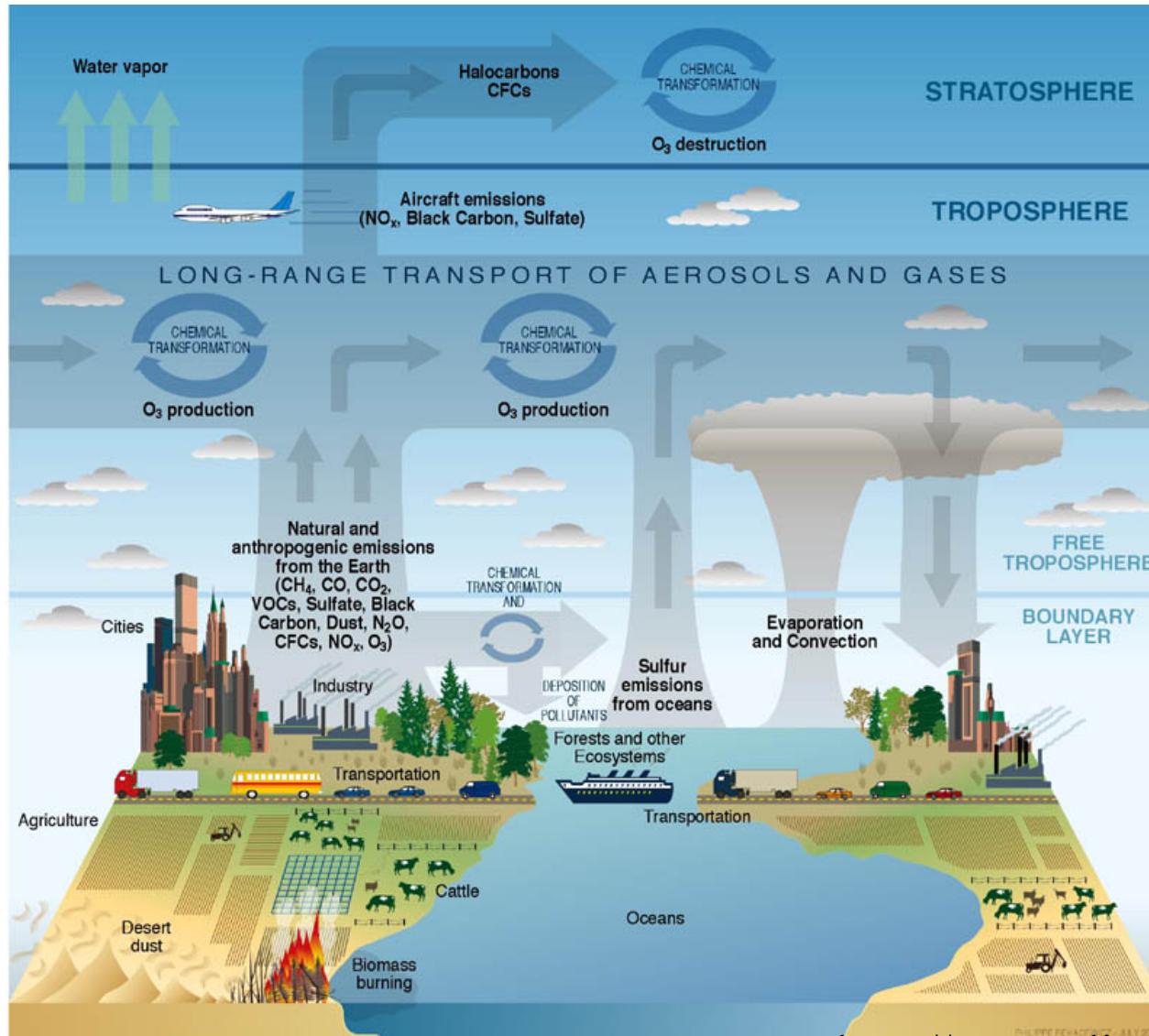


(3) Chemistry responds to changes in temperature, humidity



A.M. Fiore

Atmospheric Chemistry Processes in AM3



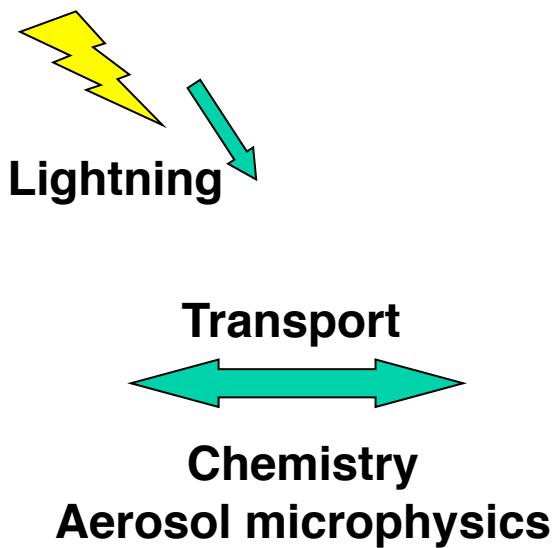
AM3 Coupled Chemistry-Climate Model

Detailed global tropospheric, stratospheric, and aerosol chemistry

- Single solver for tropospheric and stratospheric chemistry
 - 97 chemical species – 81 gases (62 transported), 16 aerosols
 - Over 200 reactions (183 kinetic, 41 photolysis)
 - MOZART-2 tropospheric chemistry (Horowitz et al., 2003) plus additional stratospheric species and reactions from AMTRAC (Austin and Wilson, 2006)
 - Sulfur chemistry fully coupled to tropospheric oxidants
 - Backward Euler solver with Newton-Raphson iteration
- Photolysis rates from multi-dimensional lookup table (TUV4.4)
- Lower boundary condition mixing ratios for methane, nitrous oxide
- Parameterized source of reactive halogens in the stratosphere
- Wet deposition: large-scale (in- and below-cloud), within convective updrafts
- Dry deposition: prescribed monthly mean deposition velocities
- CMIP5-recommended emissions (1860-2005 + 4 RCP scenarios to 2100)

Modeling Atmospheric Chemistry

Solve continuity equation for chemical mixing ratios $C_i(x, t)$



Eulerian form:

$$\frac{\partial C_i}{\partial t} = -\mathbf{U} \cdot \nabla C_i + P_i - L_i$$

Lagrangian form:

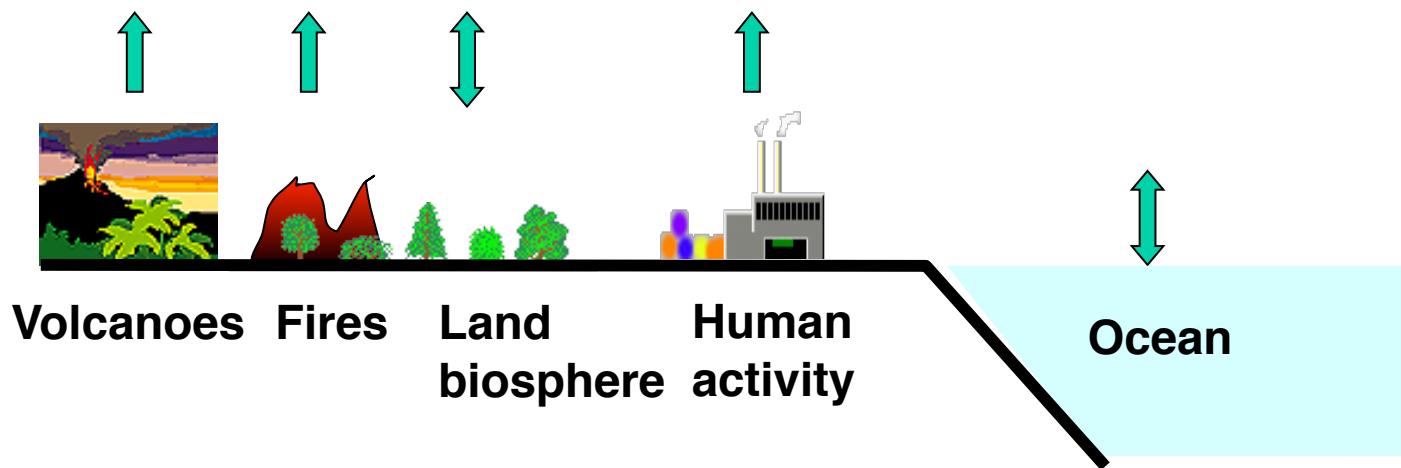
$$\frac{dC_i}{dt} = P_i - L_i$$

\mathbf{U} = wind vector

P_i = local source of chemical i

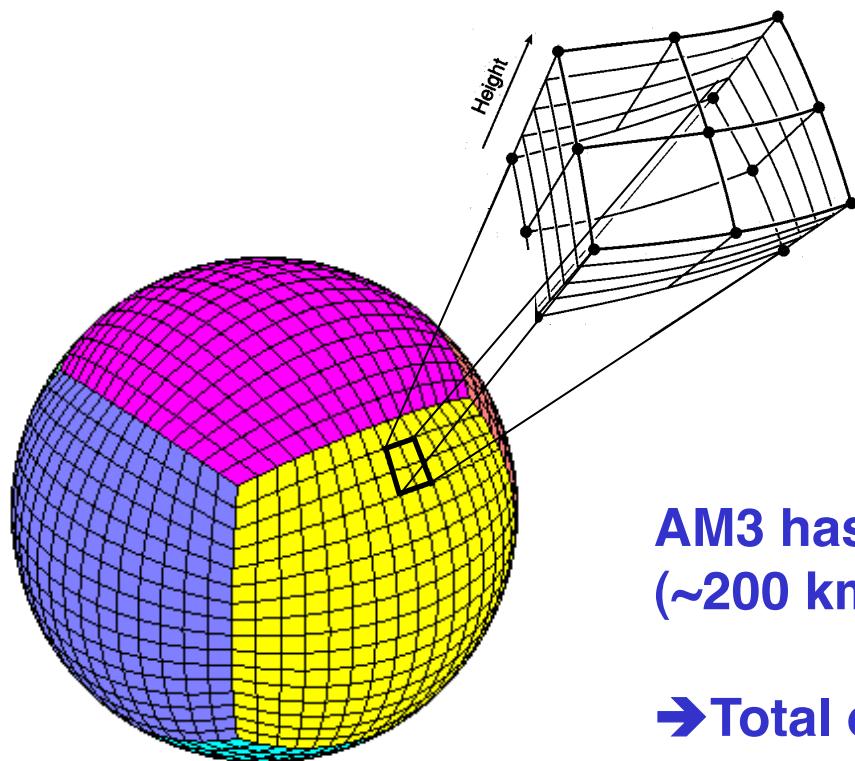
L_i = local sink

**C_i = “mixing ratio”
(mole i / mole air)**



Eulerian Models Partition Atmospheric Domain into Gridboxes

Discretize the continuity equation in space



Solve continuity equation
for individual gridboxes

AM3 has a standard resolution of c48 (~200 km), with 48 vertical levels

→ Total of ~660,000 grid cells

Chemical Composition of the Atmosphere

Mixing ratio C_i of atmospheric trace species i ($i=1,n$)

Change in concentration is determined by

- Emissions
- Deposition
- Transport at various scales

[resolved by the spatial resolution of the model and subscale (parameterisation)]

- Chemical and photochemical reactions



$$\frac{\partial C_i}{\partial t} = \left(\frac{\partial C_i}{\partial t} \right)_{emission} + \left(\frac{\partial C_i}{\partial t} \right)_{deposition} + \left(\frac{\partial C_i}{\partial t} \right)_{transport} + \left(\frac{\partial C_i}{\partial t} \right)_{chemistry}$$

Guy Brasseur

Operator Splitting in Eulerian Models

Reduces dimensionality of problem

- Split the continuity equation into contributions from transport and local terms:

$$\frac{\partial C_i}{\partial t} = \left[\frac{\partial C_i}{\partial t} \right]_{TRANSPORT} + \left[\frac{dC_i}{dt} \right]_{LOCAL}$$

Transport \equiv advection, convection: $\left[\frac{dC_i}{dt} \right]_{TRANSPORT} = -\mathbf{U} \bullet \nabla C_i$

Local \equiv chemistry, emission, deposition, aerosol processes:

$$\left[\frac{dC_i}{dt} \right]_{LOCAL} = P_i(\mathbf{C}) - L_i(\mathbf{C})$$

... and integrate each process separately over discrete time steps:

$$C_i(t_o + \Delta t) = (\text{Local}) \bullet (\text{Transport}) \bullet C_i(t_o)$$

These operators can be split further:

- split transport into 1-D advective and turbulent transport for x, y, z
- split local into chemistry, emissions, deposition

Local Chemistry Operator: solves ODE system for n interacting species

For each species $i \in [1, n]$

$$\frac{dC_i}{dt} = P_i(\mathbf{C}) - L_i(\mathbf{C}) \quad \mathbf{C} = (C_1, \dots, C_n)$$

- System is typically “stiff” (lifetimes range over many orders of magnitude)
- Simple explicit methods (e.g., forward Euler) are unstable and require the use of extremely short timesteps
- Instead, implicit methods (e.g., backward Euler) are typically used.
Transform into system of n algebraic equations with n unknowns $\mathbf{C}(t_o + \Delta t)$

$$\frac{C_i(t_o + \Delta t) - C_i(t_o)}{\Delta t} = P_i(\mathbf{C}(t_o + \Delta t)) - L_i(\mathbf{C}(t_o + \Delta t)) \quad i \in [1, n]$$

Backward Euler is stable, mass-conserving, flexible. But it is expensive (iteration required). Some 3-D models use higher-order implicit schemes such as the Gear method.

Chemical Solver

Express chemical equations as:

$$f_i(\mathbf{C}) = C_i(t_0 + \Delta t) - C_i(t_0) - \{P_i(\mathbf{C}(t_0 + \Delta t)) - L_i(\mathbf{C}(t_0 + \Delta t))\}\Delta t = 0$$

Find zero of \mathbf{f} by expanding as:

$$\mathbf{f}(\mathbf{C}) = \mathbf{f}(\mathbf{C}^0) + \frac{\partial \mathbf{f}}{\partial \mathbf{C}} (\mathbf{C} - \mathbf{C}^0) + \dots = 0$$

Express derivative in terms of Jacobian (sensitivity) matrix \mathbf{K}^0 :

$$\frac{\partial \mathbf{f}}{\partial \mathbf{C}} = I - \mathbf{K}^0 \Delta t$$

Requires
inverting

Rearrange to solve for \mathbf{C} and iterate to convergence: $n \times n$ matrix

$$\mathbf{C}^{i+1} = \mathbf{C}^i - (I - \mathbf{K}^i \Delta t)^{-1} \mathbf{f}(\mathbf{C}^i)$$

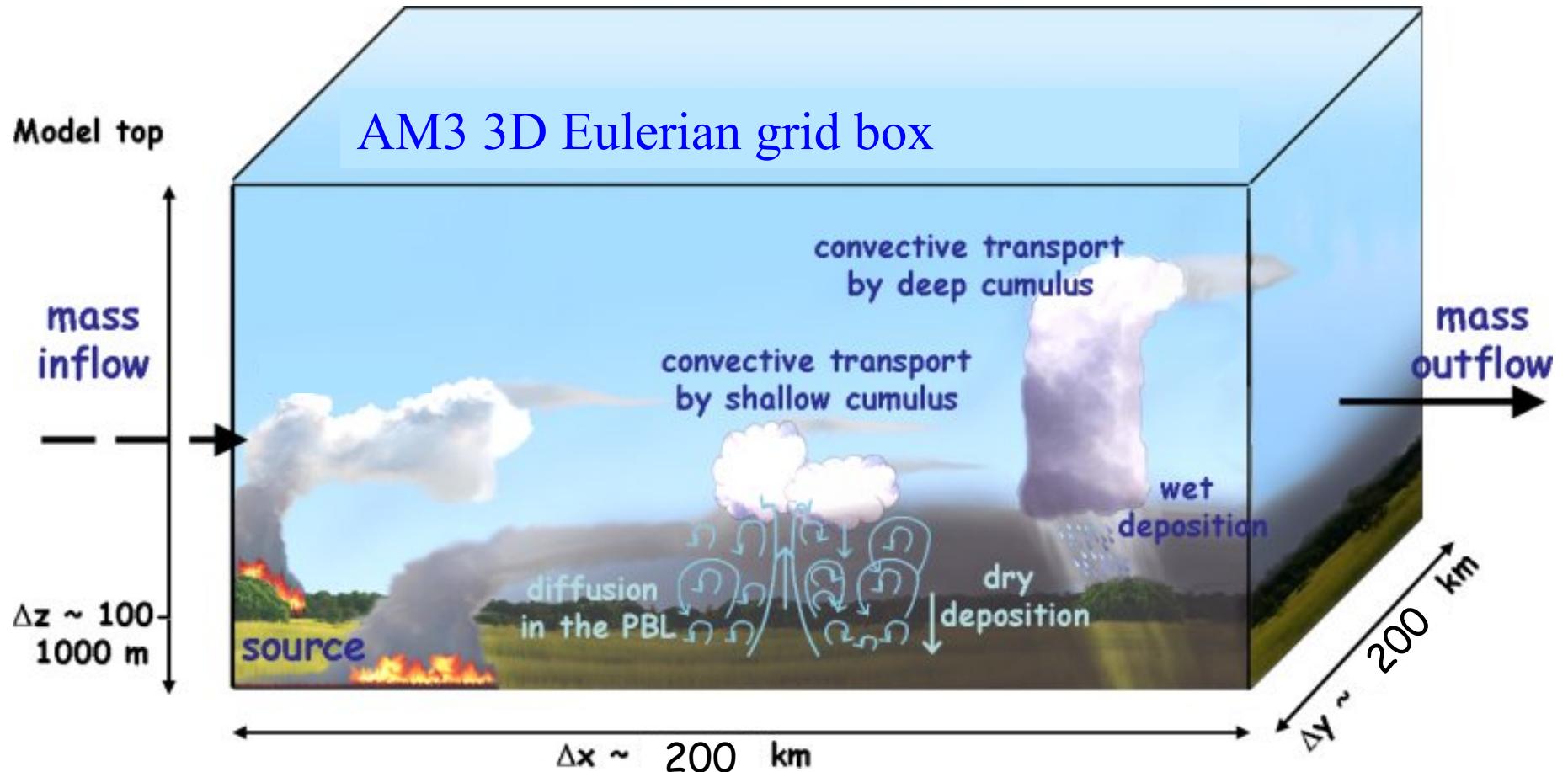
on each
iteration

AM3 requires relative convergence of 10^{-3} (or 10^{-4}).

If convergence is not achieved within 11 iterations, timestep is recursively halved up to 8 times to facilitate convergence.

Daniel Jacob

Sub-grid Processes in Gas/Aerosol Transport



Dry Deposition

Transport of gaseous and particulate species from the atmosphere onto surfaces in the absence of precipitation

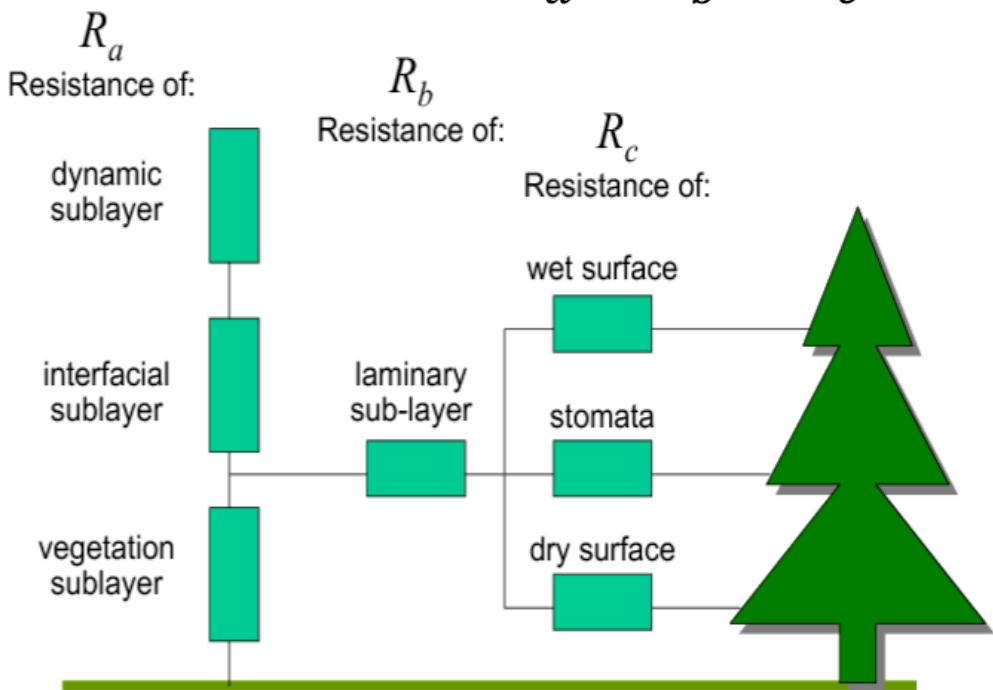
Controlling factors: atmospheric turbulence, chemical properties of species, and nature of the surface

Deposition flux: $F = -v_d C$

v_d : deposition velocity

C : concentration of species at reference height (~10 m)

$$v_d = \frac{1}{R_a + R_b + R_c}$$

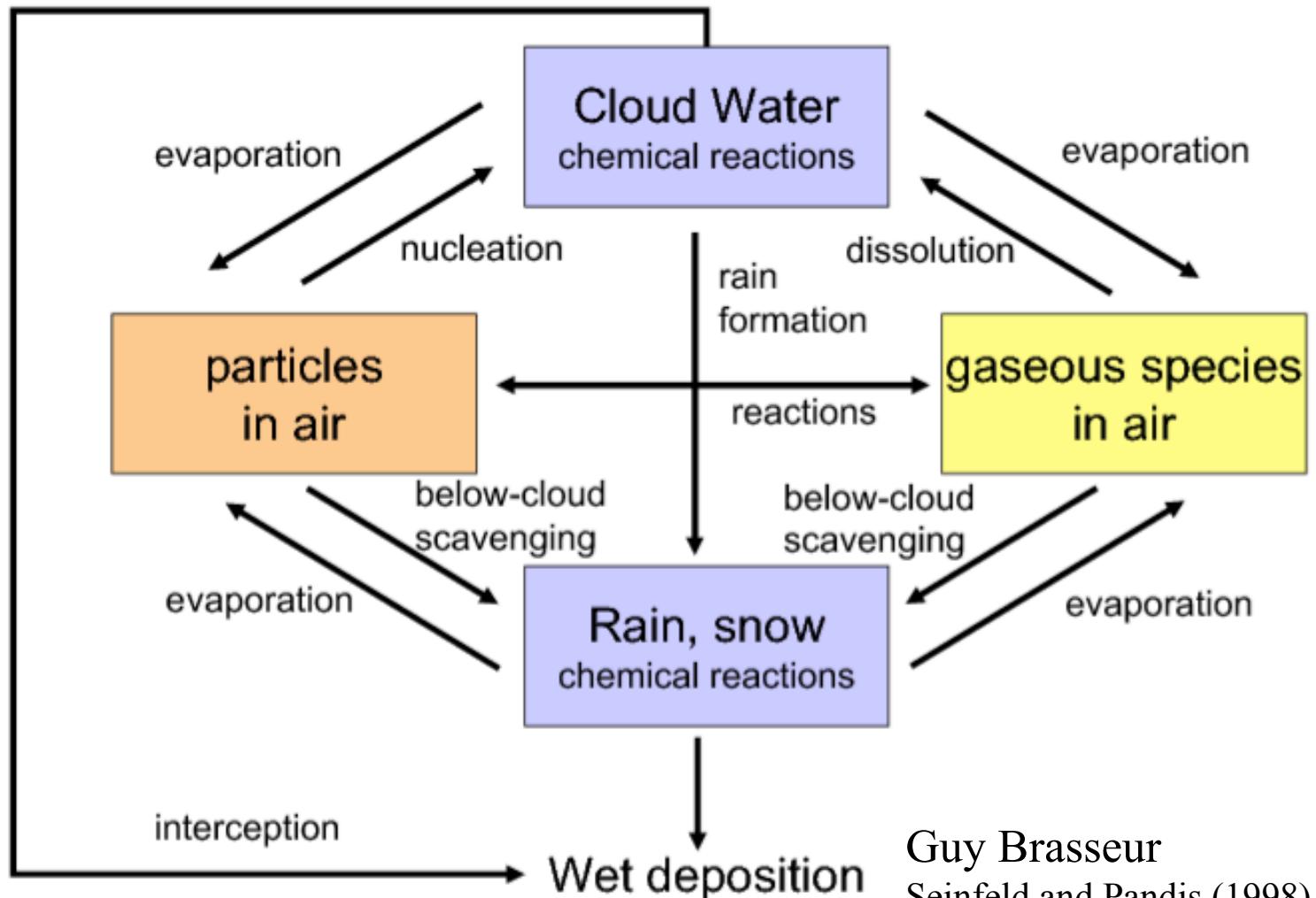


Wet Deposition

$$W = (\Gamma_{in}^{cv,ls} + \Gamma_{bc}^{ls}) \times C$$

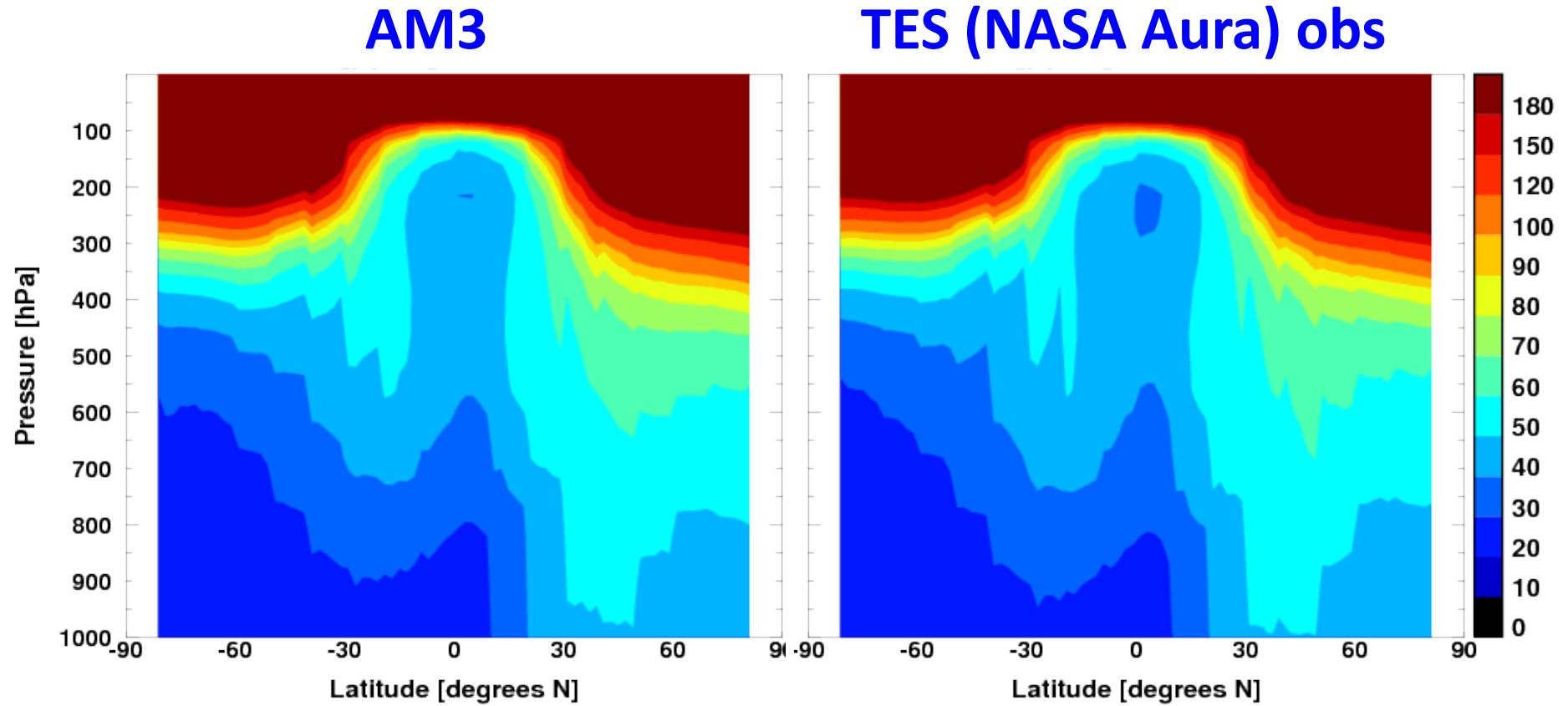
in-cloud
(convective,
large-scale)

below-cloud (large-scale only)



Guy Brasseur
Seinfeld and Pandis (1998)

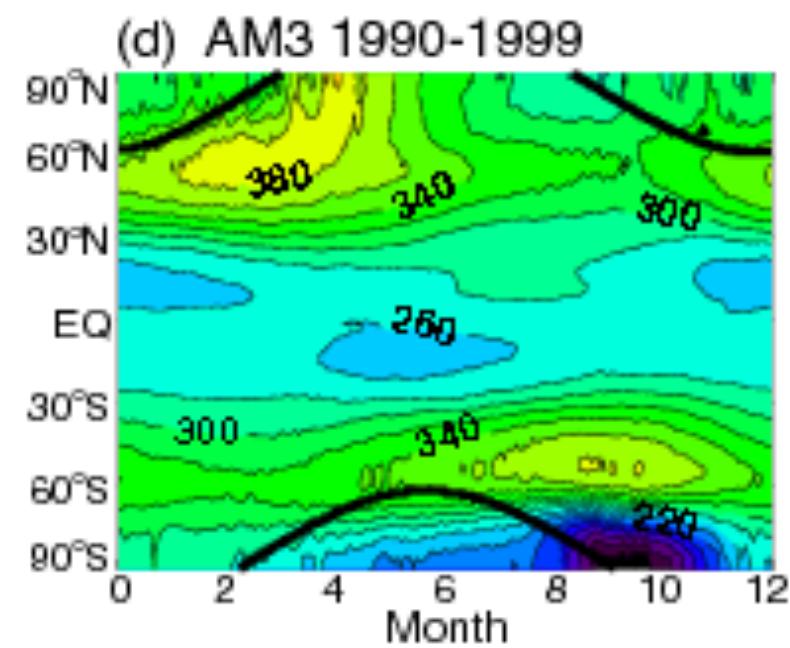
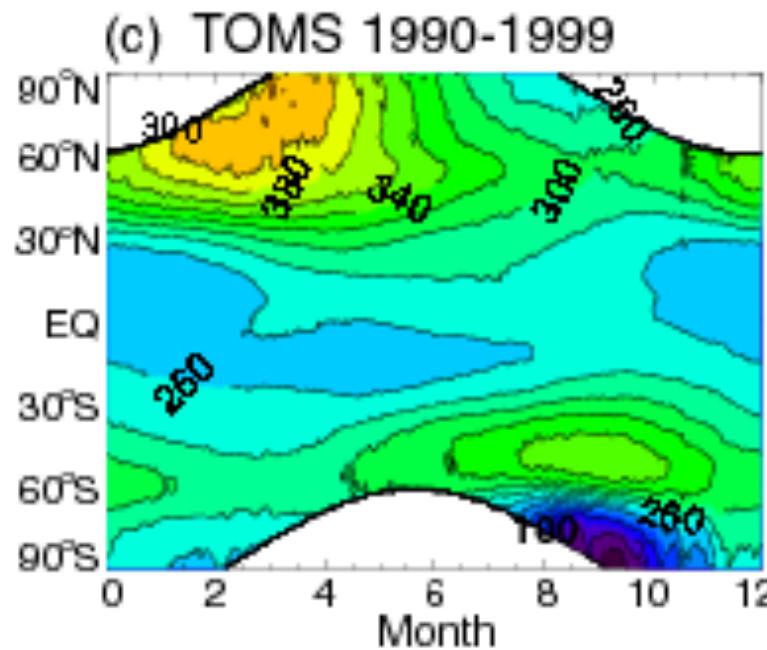
Zonal Mean Ozone (2005-2007)



AM3 accurately simulates observed magnitude and vertical/latitudinal gradients of ozone

Vaishali Naik *et al.* (2012, submitted)

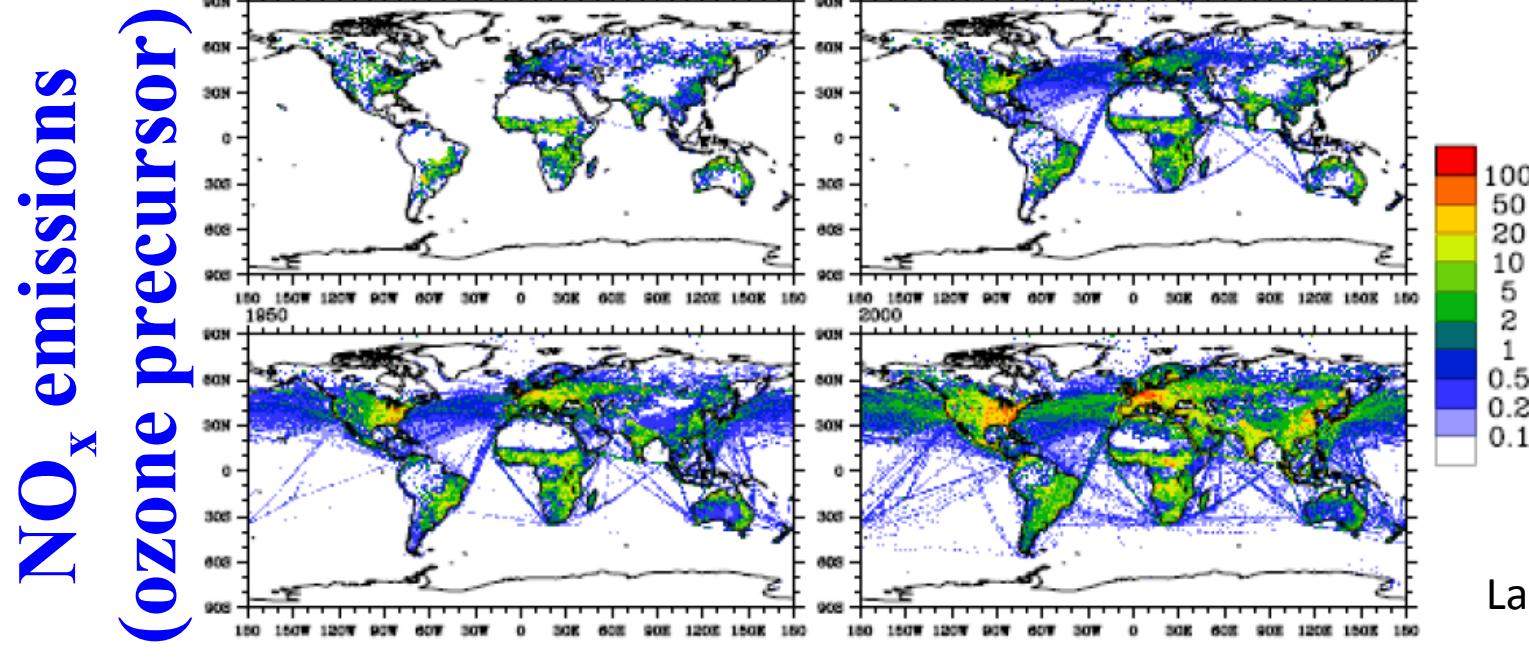
Ozone Column (Dobson Units)



AM3 simulates major features of observed seasonal and latitudinal ozone column

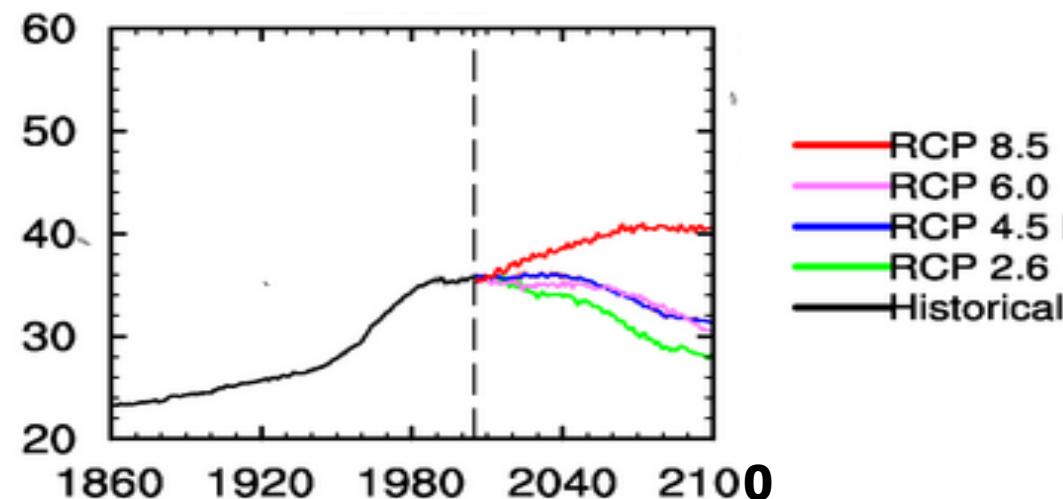
Donner *et al.*, *J. Climate*, 2011

Significant anthropogenic perturbation to atmospheric chemistry



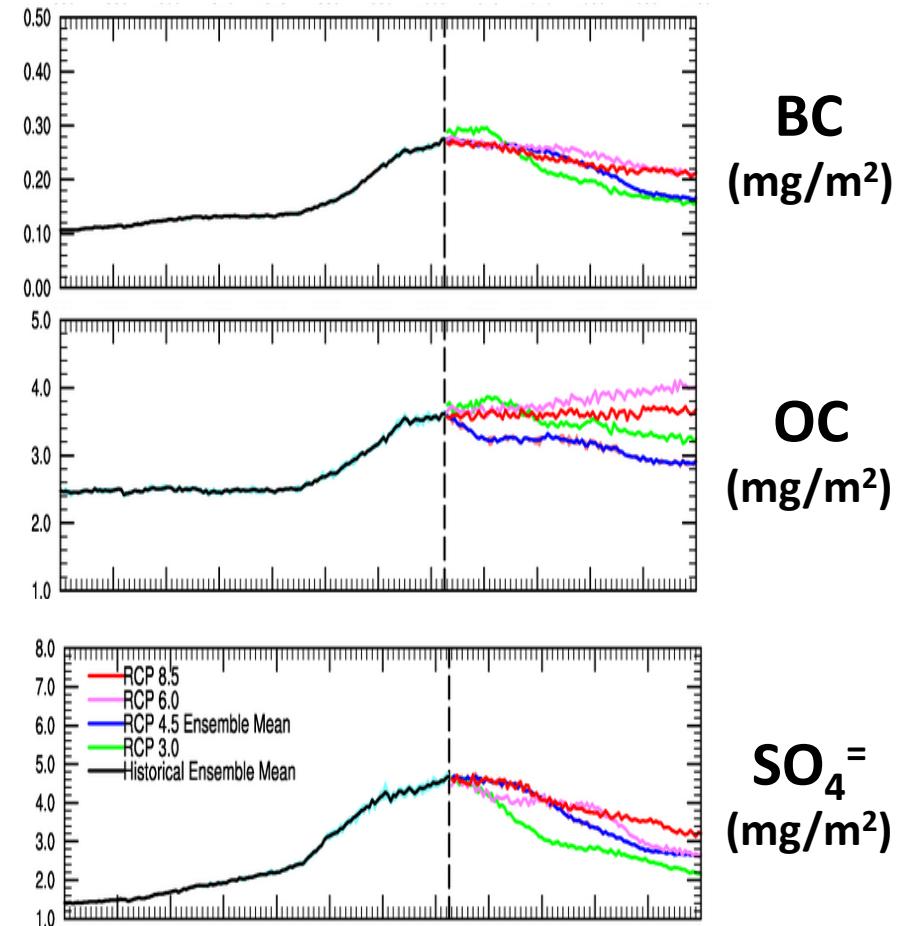
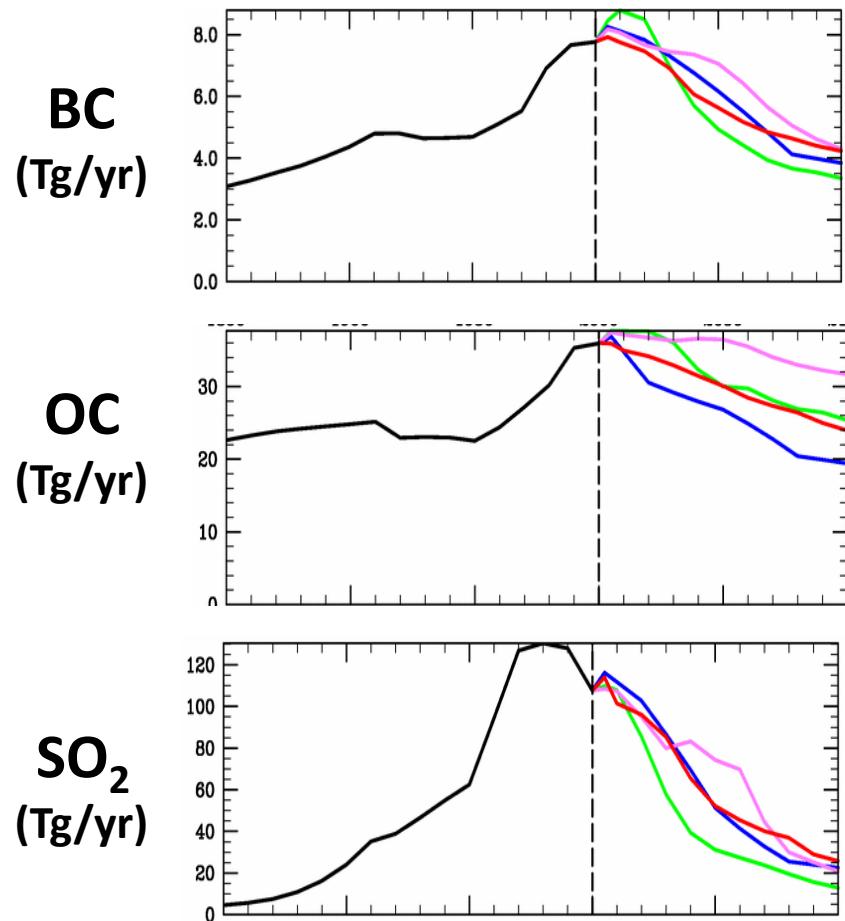
Lamarque et al.
(2010)

Global mean
surface O₃
concentration



Vaishali Naik

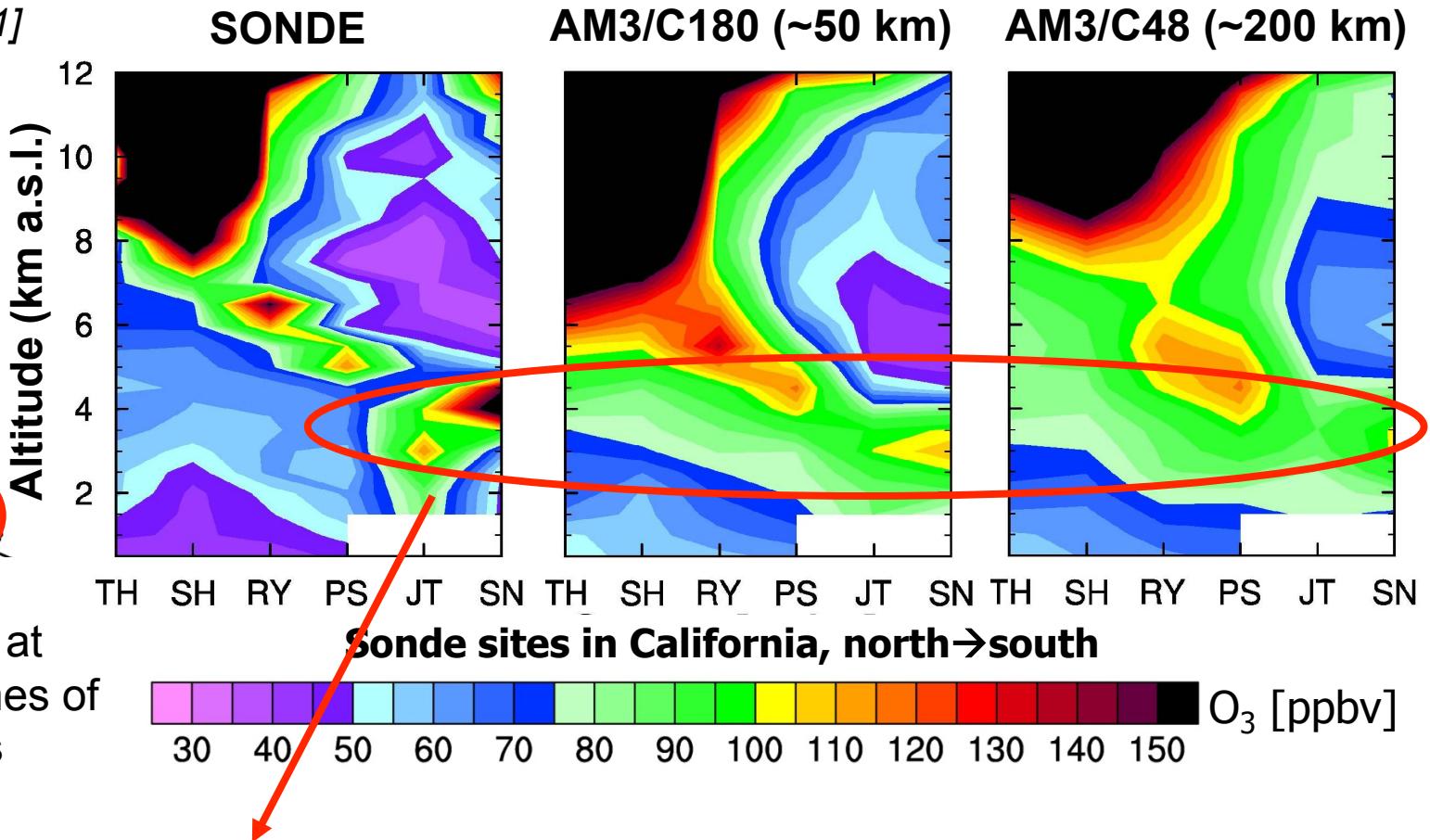
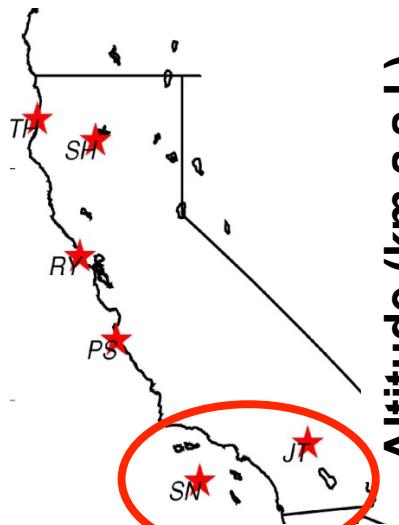
Aerosol Emissions and Burdens



Vaishali Naik

High-resolution (c180, 50 km) “nudged” version of AM3: CalNex case study (May 28, 2010)

Cooper et al. [2011]



model sampled at
location and times of
sonde launches

- High O₃ mixing ratios (90-150 ppbv) just 2-4 km aloft populated LA
- AM3/C48 reproduces the large-scale view
- AM3/C180 better captures vertical structure

Meiyun Lin et al., 2012, submitted

Configuring Chemistry in AM3: namelists

```
<namelist name="tropchem_driver_nml">
    relaxed_dt = 864000.,                      # timescale (s) for upper boundary condition relaxation
    ub_pres = 1.,                                # pressure (Pa) below which to apply upper BC
    relaxed_dt_lbc = 86400.,                     # timescale (s) for lower boundary condition relaxation
    lb_pres = 800.e2,                            # pressure (Pa) above which to apply lower BC
    file_sulfate = 'sulfate.nc',      # file for climatological sulfate concentration (not used)
    file_conc = 'tracerIC.19800101.v2.nc', # IC for tracers (if not included in std IC file)
    file_emis_1 = 'emissions.',      # String used in constructing default emission filename:
    file_emis_2 = '.1x1.1859_2020.nc',          #     $file_emis_1.$species.$file_emis_2
    file_emis3d_1 = 'emissions3D.',            # String used in constructing default 3D emission filename:
    file_emis3d_2 = '.1x1.1859_2020.nc',          #     $file_emis3d_1.$species.$file_emis3d_2
    file_ub = 'ub_vals.mozart.nc',             # filename for upper boundary condition
    file_dry = 'depvel.nc',                    # filename for dry deposition velocities
    inv_list = '',                           # "invariant" species list (besides M, O2, N2, {H2O})
    file_aircraft = 'emissions.aircraft.1x1.1859_2020.nc', # filename for aircraft emissions
    light_no_prd_factor = 0.6,                 # scale factor for lightning NOx (function of convection)
    strat_chem_age_factor = 1.25,              # scale factor for stratospheric "age of air"
    strat_chem_dcliydt_factor = 1.,           # scale factor for stratospheric Cly source
    do_tropchem = .true.,                     # activate chemistry module?
    do_fastjx_photo = .false.,                # calculate photolysis interactively? (F=lookup table)
    use_tdep_jvals = .true.,                  # use T-dependent entries in photolysis lookup table?
    file_jval_lut = 'jvals.v6fix.twilight.solarmax', # photolysis LUT file (solar max)
    file_jval_lut_min = 'jvals.v6fix.twilight.solarmin', # photolysis LUT file (solar min)
    o3_column_top = 0.002,                   # ozone column above model top
    repartition_water_tracers = .true.,      # partition cloud+gas water using strat aerosol/PSC scheme
    allow_negative_cosz = .true.,            # allow twilight photolysis?
    allow_psc_settling_type1 = .true.,       # allow settling of ice PSCs?
    force_cly_conservation = .true.,         # force solver to exactly conserve total Cly?
    set_min_h2o_strat = .true.,              # set floor for strat. H2O (2*CH4, for chemistry only)?
    ch4_filename = 'ch4_gblannualdata',      # CH4 timeseries file
    ch4_scale_factor = 1.e-9,                # CH4 timeseries unit conversion (ppb→VMR)
    verbose = 1,                            # control verbosity of diagnostic messages
    retain_cm3_bugs = .false.,              # for backward compatibility
    check_convergence = .true.               # discard chemical tendencies if solver not converged?
</namelist>
```



Configuring Chemistry in AM3 : carbon aerosol namelist

```
<namelist name="carbon_aerosol_nml">
  bcff_source = 'gocart_2007',                                # Black carbon fossil fuel source
  bcff_input_name(1)='anthro',                                 # field name in file for BC FF
  bcff_filename='emissions.bc.1x1.1859_2020.nc', # filename for BC fossil fuel source
  bcff_time_dependency_type='time_varying',bcff_dataset_entry=$baseDate # time dependence for BC FF
  omff_source = 'gocart_2007',                                # Organic carbon fossil fuel source
  omff_input_name(1)='anthro',
  omff_filename='emissions.om.1x1.1859_2020.nc',
  omff_time_dependency_type='time_varying',omff_dataset_entry=$baseDate
  omsh_source = 'gocart_2007',                                # Organic carbon source from ships
  omsh_input_name(1)='ship',
  omsh_filename='emissions.om.1x1.1859_2020.nc',
  omsh_time_dependency_type='time_varying',omsh_dataset_entry=$baseDate
  bcsh_source = 'gocart_2007',                                # Black carbon source from ships
  bcsh_input_name(1)='ship',
  bcsh_filename='emissions.bc.1x1.1859_2020.nc',
  bcsh_time_dependency_type='time_varying',bcsh_dataset_entry=$baseDate
  bcav_source = 'do_aircraft'                                 # Black carbon source from aircraft
  bcav_input_name(1)='fuel',
  bcav_filename='emissions.aircraft.aero.1x1.1859_2020.nc',
  bcav_time_dependency_type='time_varying',bcav_dataset_entry=$baseDate
  bcbb_source = 'AEROCOM',                                    # Black carbon biomass burning source
  bcbb_filename='emissions.bc.1x1.1859_2020.nc',
  bcbb_input_name(1)='bb_11'
  bcbb_input_name(2)='bb_12'
  bcbb_input_name(3)='bb_13'
  bcbb_input_name(4)='bb_14'
  bcbb_input_name(5)='bb_15'
  bcbb_input_name(6)='bb_16'
  bcbb_time_dependency_type='time_varying',bcbb_dataset_entry=$baseDate
  ombb_source = 'AEROCOM',                                    # Organic carbon biomass burning source
  ombb_filename='emissions.om.1x1.1859_2020.nc',
  ombb_input_name(1)='bb_11'
  ombb_input_name(2)='bb_12'
  ombb_input_name(3)='bb_13'
  ombb_input_name(4)='bb_14'
  ombb_input_name(5)='bb_15'
  ombb_input_name(6)='bb_16'
  ombb_time_dependency_type='time_varying',ombb_dataset_entry=$baseDate
  omna_source = 'gunther',                                     # Organic carbon natural source (scaled to terpene emissions)
  omna_filename='oc8_natural_emission.nc'
  omna_input_name='omemisnat'
  omss_source = 'ODowd'                                       # Organic carbon source from ocean
  omss_coef = 0.05
</namelist>
```

Configuring Chemistry in AM3 : hygroscopic growth

```
<namelist name="aerosolrad_package_nml">
  using_im_bcsul = .true.                                # use sulfate/BC internal mixture?
  volcanic_dataset_entry = $baseDate                   # date for stratospheric (volcanic) aerosol file
  using_volcanic_lw_files = .true.,                     # do stratospheric aerosols have LW effect?
  using_volcanic_sw_files = .true.,                     # do stratospheric aerosols have SW effect?
  do_lwaerosol = .true.,                               # do aerosols have LW effect?
  do_swaerosol = .true.,                               # do aerosols have SW effect?
  aerosol_data_set = "Ginoux_Rddy"                    # aerosol optical property dataset name
  optical_filename = "Ginoux_Rddy_2005",               # aerosol optical property dataset filename
  sulfate_indices = 30,30,30,30,30,30,30,30,30,30,30,
                  30,30,30,30,30,30,30,30,30,30,30,30,
                  30,35,35,35,35,35,40,40,40,40,40,45,45,45,45,
                  50,50,50,50,50,55,55,55,55,55,60,60,60,60,60,65,
                  65,65,65,65,70,70,70,70,75,75,75,75,75,80,80,
                  80,80,82,82,84,84,86,86,88,88,90,91,92,93,94,95,
                  96,97,97,97,97,
...
</namelist>
```

sulfate_indices(0:100)
set RH for hygroscopic
growth for given
ambient RH
→ growth to 97% RH

Change to:

```
sulfate_indices = 30,30,30,30,30,30,30,30,30,30,30,
                  30,30,30,30,30,30,30,30,30,30,30,
                  30,35,35,35,35,35,40,40,40,40,40,45,45,45,45,
                  50,50,50,50,50,55,55,55,55,55,60,60,60,60,60,65,
                  65,65,65,65,70,70,70,70,75,75,75,75,75,80,80,
                  80,80,82,82,84,84,86,86,88,88,90,90,90,90,90,90,
```

→ growth to 90% RH

Configuring Chemistry in AM3 : field table

field_table_am3p8_o3

```
"tracer", "atmos_mod", "so2" # define atmospheric tracer
"longname", "SO2"           # tracer long name
"units", "VMR"              # tracer units (volume mixing ratio)
"emissions", "file"         # read in emissions from default file (tropchem_driver_nml)
"emissions3d", "file", "file=emissions3d.SO2.2000.v4plev.nc" # from specified file
"profile_type", "fixed", "surface_value=1.e-20" # if no IC, set to fixed value
"wet_deposition", "henry_below", "henry=1.49e-2, dependence=5080." # wet deposition
"dry_deposition", "fixed", "land=0.25e-2, sea=0.6e-2" # dry deposition
"aircraft_emis", "file" # aircraft emissions from default file (see tropchem_driver_nml)
"convection", "all" / # include convective transport

"tracer", "atmos_mod", "so4" # define atmospheric tracer
"longname", "SO4"           # tracer long name
"units", "VMR"              # tracer units (volume mixing ratio)
"profile_type", "fixed", "surface_value=1.e-20" # if no IC, set to fixed value
"wet_deposition", "aerosol_below", # wet deposition
  "frac_incloud=0.3,           # scavenging efficiency in LS clouds
  frac_incloud_uw=0.4, frac_incloud_donne=0.4, # scav eff. In convective clouds
  alphar=0.001, alphas=0.001"          # below-cloud scav. efficiency
"dry_deposition", "fixed", "land=0.11e-2, sea=0.09e-2" # prescribed dry dep. velocity
"convection", "all"           # allow convective transport
"radiative_param", "online", "name_in_rad_mod=so4,      # species radiatively active
                      name_in_clim_mod=sulfate, # name in optics file
                      scale_factor=4.56" /    # units conversion (VMR→MMR)
```

Chemistry Options in AM3: emissions files

```
<dataFile label="input" target="INPUT/" chksum="" size="" timestamp="">
    <dataSource platform="${platform}">
        ${FMS_ARCHIVE_ROOT}/cm3/ipcc_ar5/input/historical/emissions.1x1.1859_2020.tar
    </dataSource>
</dataFile>
```

```
> tar -tf emissions.1x1.1859_2020.tar
emissions3D.acet.1x1.1859_2020.nc
emissions3D.c2h4.1x1.1859_2020.nc
emissions3D.c2h5oh.1x1.1859_2020.nc
emissions3D.c2h6.1x1.1859_2020.nc
emissions3D.c3h6.1x1.1859_2020.nc
emissions3D.c3h8.1x1.1859_2020.nc
emissions3D.c4h10.1x1.1859_2020.nc
emissions3D.ch2o.1x1.1859_2020.nc
emissions3D.ch3oh.1x1.1859_2020.nc
emissions3D.co.1x1.1859_2020.nc
emissions3D.h2.1x1.1859_2020.nc
emissions3D.isoprene.1x1.1859_2020.nc
emissions3D.nh3.1x1.1859_2020.nc
emissions3D.no.1x1.1859_2020.nc
emissions3D.terpenes.1x1.1859_2020.nc
emissions.acet.1x1.1859_2020.nc
emissions.aircraft.1x1.1859_2020.nc
emissions.aircraft.aero.1x1.1859_2020.nc
emissions.bc.1x1.1859_2020.nc
emissions.c2h4.1x1.1859_2020.nc
emissions.c2h5oh.1x1.1859_2020.nc
emissions.c2h6.1x1.1859_2020.nc
emissions.c3h6.1x1.1859_2020.nc
emissions.c3h8.1x1.1859_2020.nc
emissions.c4h10.1x1.1859_2020.nc
emissions.ch2o.1x1.1859_2020.nc
emissions.ch3oh.1x1.1859_2020.nc
emissions.co.1x1.1859_2020.nc
emissions.h2.1x1.1859_2020.nc
emissions.isoprene.1x1.1859_2020.nc
emissions.nh3.1x1.1859_2020.nc
emissions.no.1x1.1859_2020.nc
emissions.om.1x1.1859_2020.nc
emissions.so2.1x1.1859_2020.nc
emissions.terpenes.1x1.1859_2020.nc
```

```
> ncdump -h emissions.so2.1x1.1859_2020.nc
netcdf emissions.so2.1x1.1859_2020{
dimensions:
    lon = 360 ;
    lat = 180 ;
    time = UNLIMITED ; // (228 currently)
variables:
    float lon(lon) ;
        lon:long_name = "longitude" ;
        lon:units = "degrees_east" ;
    float lat(lat) ;
        lat:long_name = "latitude" ;
        lat:units = "degrees_north" ;
    float time(time) ;
        time:standard_name = "time" ;
        time:long_name = "time" ;
        time:units = "days since 1850-01-01 00:00:00" ;
        time:calendar = "julian" ;
    float anthro(time, lat, lon) ;
        anthro:history = "SO2:emiss_awb+SO2:emiss_dom+
                           SO2:emiss_enet+SO2:emiss_ind+
                           SO2:emiss_tra+SO2:emiss_wst" ;
        anthro:long_name = "anthropogenic emissions
                           (incl. biofuels)" ;
        anthro:units = "molecules/cm2/s" ;
    float bb(time, lat, lon) ;
        bb:history = "SO2:grassfire+SO2:forestfire" ;
        bb:long_name = "biomass burning emissions" ;
        bb:units = "molecules/cm2/s" ;
    float ship(time, lat, lon) ;
        ship:history = "SO2:emiss_shp" ;
        ship:long_name = "shipping emissions" ;
        ship:units = "molecules/cm2/s" ;
// global attributes:
}
```

Configuring Chemistry in AM3 : chemical pre-processor

Solution species

```
( 1) O3  
( 2) O  
( 3) O1D      (O)  
( 4) N2O
```

Photolysis

```
jo2   ( 1) O2 + hv -> 2*O  
jo1d  ( 2) O3 + hv -> O1D + O2  
jo3p  ( 3) O3 + hv -> O + O2  
jn2o  ( 4) N2O + hv -> O1D + N2
```

Provide list of species, reactions, rates

```
rate = ** User defined **      ( 1)  
rate = ** User defined **      ( 2)  
rate = ** User defined **      ( 3)  
rate = ** User defined **      ( 4)
```

Reactions

```
usr1   ( 1) O + O2 + M -> O3 + M  
       ( 2) O + O3 -> 2*O2  
ox_p1  ( 8) NO + HO2 -> NO2 + OH  
       ( 9) NO + O3 -> NO2 + O2  
       (10) NO2 + O -> NO + O2  
usr2   (13) NO2 + NO3 + M -> N2O5 + M
```

```
rate = ** User defined **      ( 42)  
rate = 8.00E-12*exp( -2060./t) ( 43)  
rate = 3.50E-12*exp( 250./t)   ( 49)  
rate = 3.00E-12*exp( -1500./t) ( 50)  
rate = 5.10E-12*exp( 210./t)   ( 51)  
troe : ko=2.00E-30*(300/t)**4.40 ( 54)  
      ki=1.40E-12*(300/t)**0.70  
      f=0.60
```

... generates Fortran code for solving chemical system (LU-decomposition sparse matrix inversion):

```
subroutine imp_lu_fac01( lu )  
use CHEM_MODS_MOD, only : imp_nzcnt, clsze  
implicit none  
real, intent(inout) :: lu(imp_nzcnt)  
  
lu(1) = 1. / lu(1)  
lu(2) = 1. / lu(2)  
lu(3) = 1. / lu(3)  
lu(4) = 1. / lu(4)  
lu(6) = 1. / lu(6)  
lu(7) = lu(7) * lu(6)  
lu(801) = lu(801) - lu(7) * lu(743)
```